Sandia National Laboratories

Exceptional service in the national interest

# ME469: Nalu Overview

Stefan P. Domino[1,2]

[1] Computational Thermal and Fluid Mechanics, Sandia National Laboratories

[2] Institute for Computational and Mathematical Engineering, Stanford



SAND2018-4536 PE

U.S. DEPARTMENT OF ENERGY

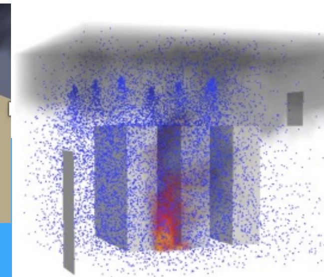
NNSA

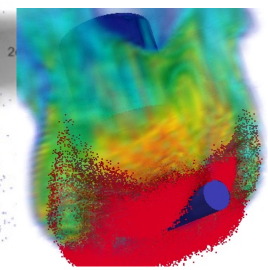
# Lecture Objectives

- Nalu Technology Origination: Advanced Simulation and Computing project (NNSA)
- Beyond the 32-bit Limit
- Supported Physics
- Supported Numerics
- Low- and High-order
- Moving Mesh (Sliding and Overset)
- Multiphysics
  - Fluid Structure Interaction
  - Conjugate Heat Transfer (CHT)
  - Participating Media Radiation
  - Examples

# Core Technology Provided to Nalu Origination: Advanced Simulation and Computing Sierra/Fuego

- Use-case characterized by a highly sooting, turbulent, reacting flow with Participating Media Radiation (PMR), Conjugate Heat Transfer (CHT), and propellant multi-physics coupling

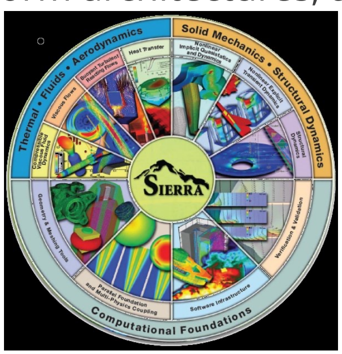




- Complex geometry has driven a generalized, hybrid unstructured discretization approach supporting Hex8, Tet4, Wedge6, and Pyramid5 elements in addition to promotion of Hex8 to Hex27, and Tet4 to Tet10

## Goal: Beyond 32-bit Computing

- Circa 2013, many scientific production codes were limited to 32-bit
- Therefore, maximum simulation size for entities, e.g., node, edge, face, element, etc., was ~2.2 billion
- Next Generation Platforms were advocated to overcome poor MPI scaling and power needs to support Exascale computing ($10^{18}$ floating point operations/second)
    - Platform architectures, at that point, were not not yet known (still evolving)




Trilinos

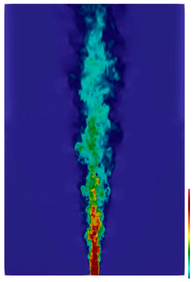
+ ASC Investments

(  )

Sierra Toolkit/Trilinos (open-source)
MPI+X parallelism
Support for new architectures

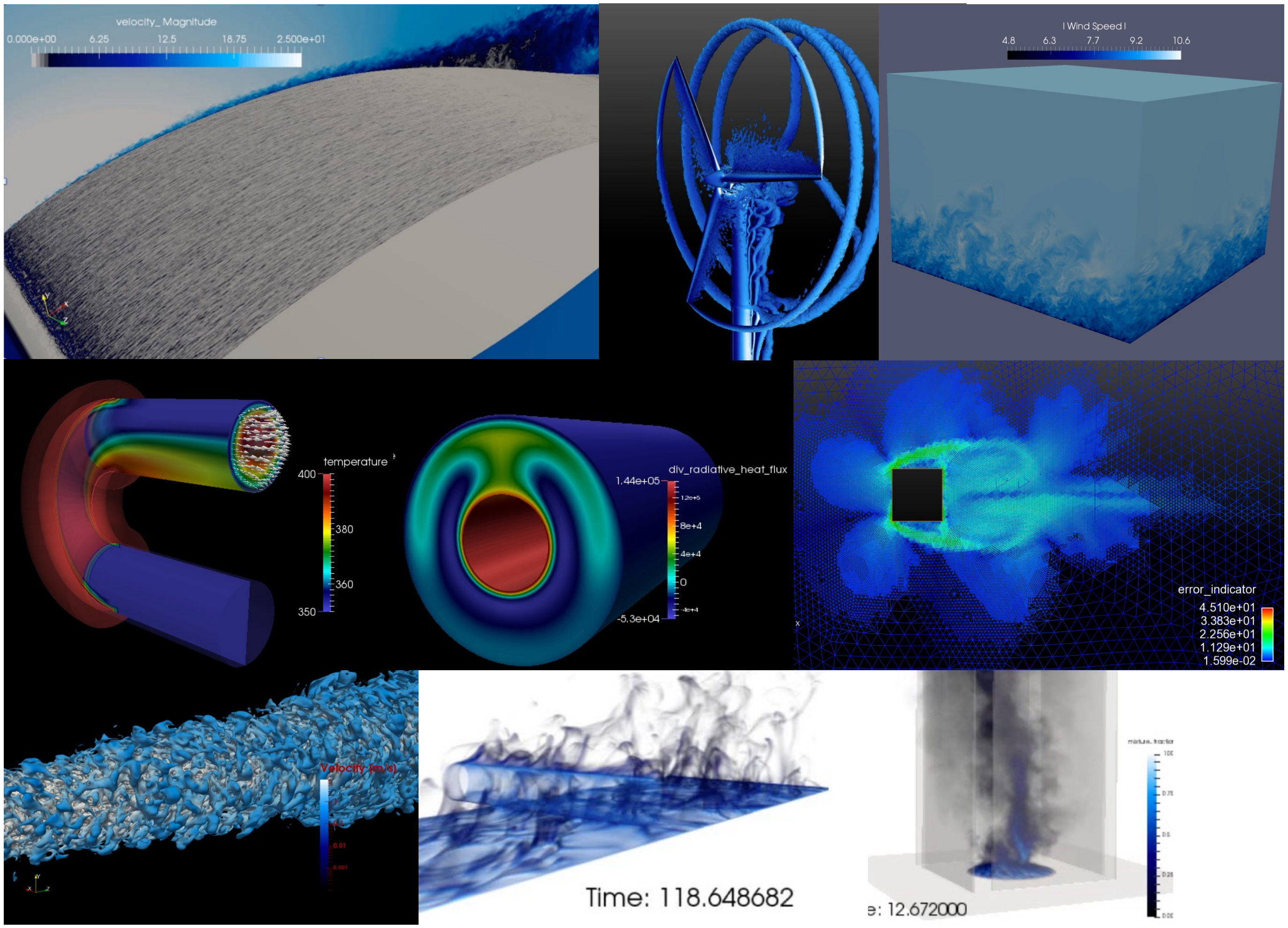
# Developed Open-Source BSD-clause 3 Distribution Policy

- Philosophy: Open-source collaborations



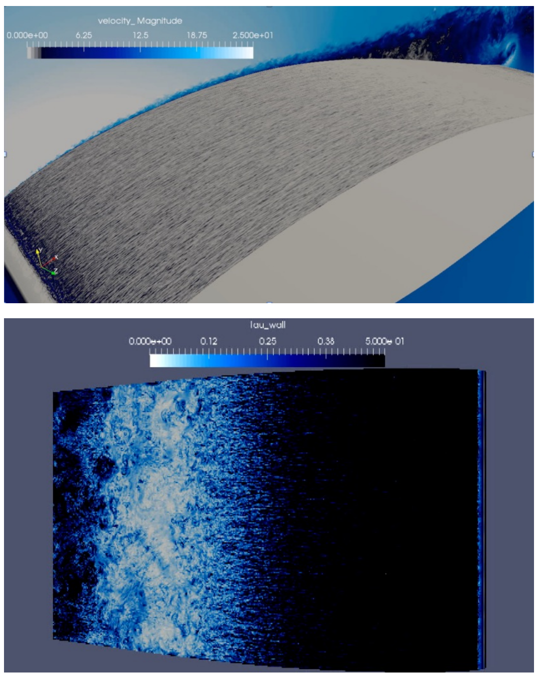
https://github.com/NaluCFD

# Supported Physics: DNS and LES (even RANS)

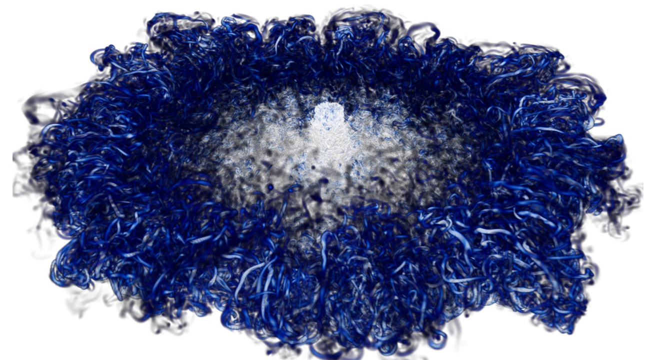


# A Note on High Performance Computing

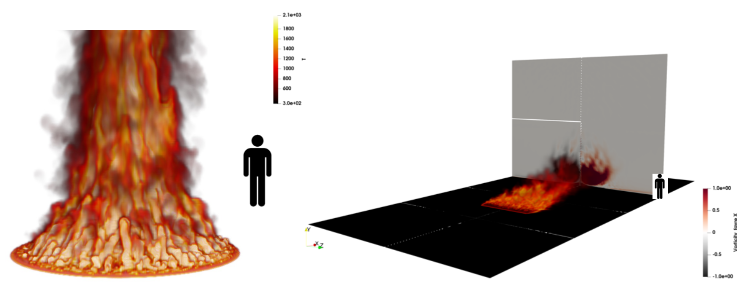
Sandia (and Stanford) are committed to High Performance Computing (HPC) to support is science and engineering objectives: Exercised herein




O(6) billion wind energy application



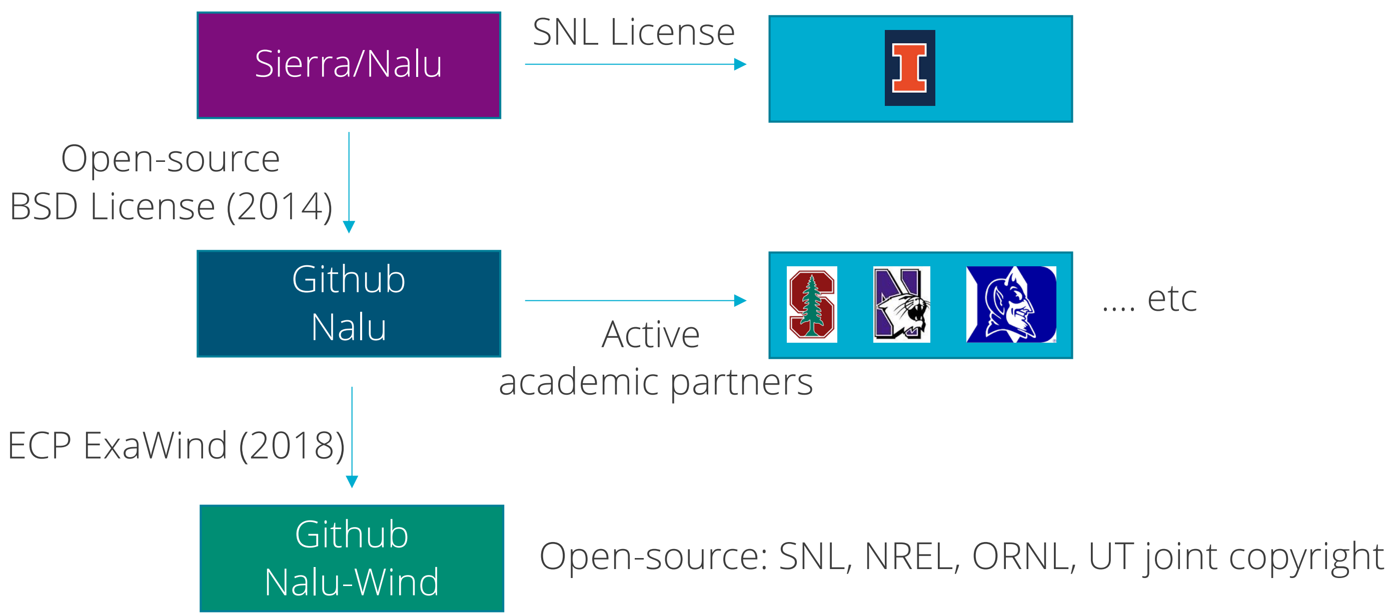
O(2) billion DNS impinging jet



O(200) million multi-physics fire

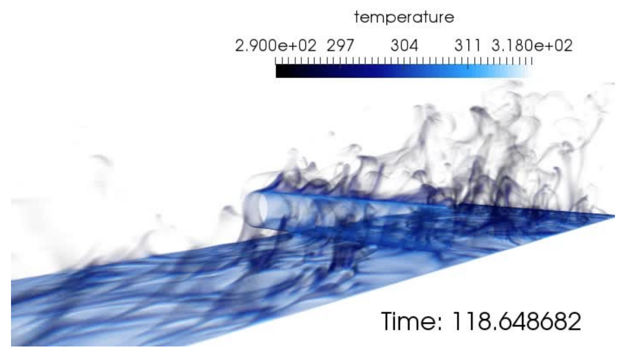
# Nalu Timeline/History

- By CFD standards, this is a relatively new code base

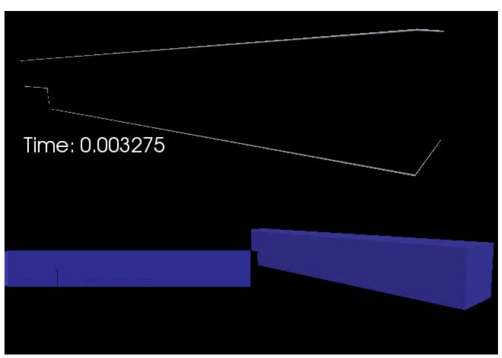


# Several Multi-physics Flow Examples From Nalu

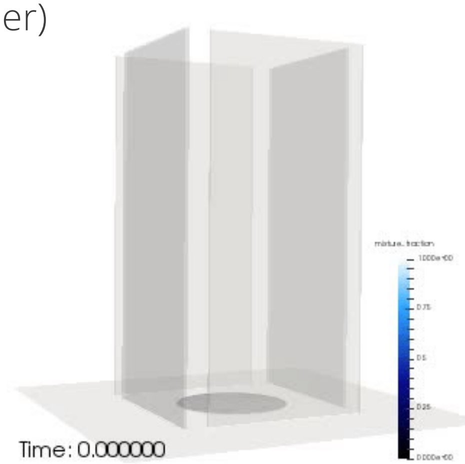

Helium plume (low-/high-order)



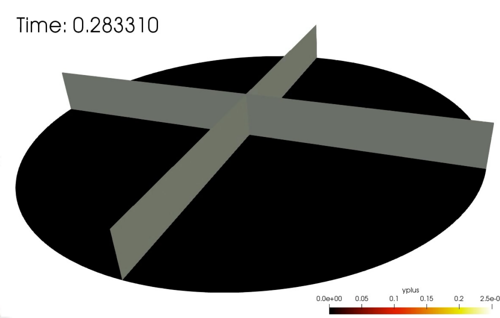

Cylinder in cross flow




Heated backstep

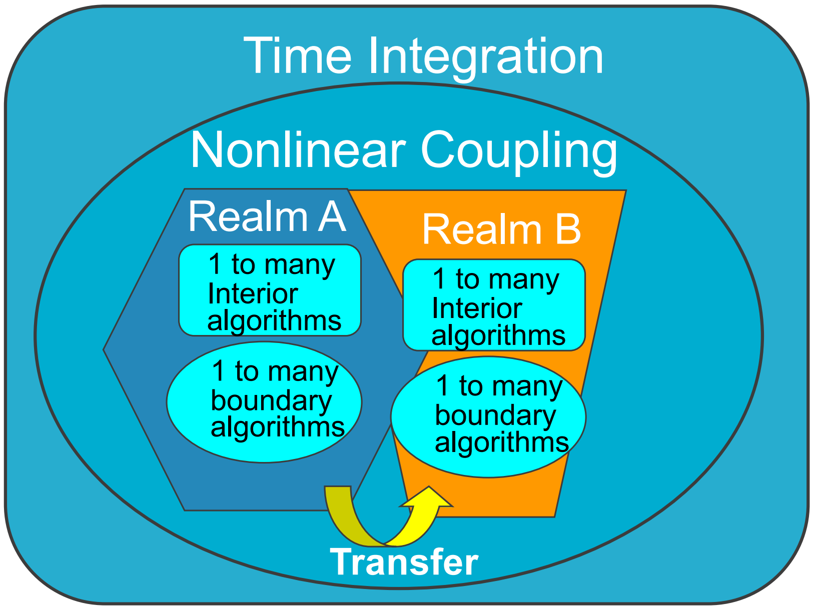



Whirling behavior



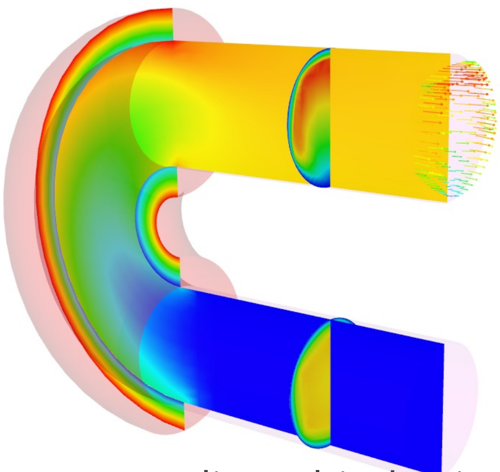

Impinging Jet DNS

# Nalu Abstractions: Polymorphic and Input File-Driven




- *Realm* specifications define the physics and desired boundary conditions
- Pre-defined *EquationSystems* (segregated or monolithic)

Operator-split multi-physics
Conjugate heat transfer coupling
- Fluids Realm
- Heat Conduction Realm



Operator-split multi-physics:

- Conjugate heat transfer coupling
  - Fluids and solid Realm

# 30K View:  Anatomy of a Nalu Input File:YAML-based

Simulation:

linear_solvers: Specification of sparse Trilinos-based precond/solver

transfers: Data transfer for multi-physics coupling

realms:

*YAML enforces strict spacing and ordering*

https://www.democraticunderground.com/10021540110

```
  - name: realm_heatCond

    boundary_conditions:
    - wall_boundary_condition: bc_exposed

    solution_options:
    initial_conditions:
    material_properties:
    equation_systems:

      systems:
        - HeatConduction:

    output:
    restart:

  - name: realm_fluids
```

Physics definitions

TimeIntegrators: Time integration, e.g., BE, BDF2

# High-Level Elements of an Input File

```
systems:

  - LowMachEOM:
      name: myLowMach

  - MixtureFraction:
      name: myZ
```

```
initial_conditions:

  - constant: ic_1
      target_name: [block_1, ...]
      value:
        pressure: 0
        velocity: [0.5,0.0]
        mixture_fraction: 0.0
```

```
boundary_conditions:

  - inflow_boundary_condition: bc_left_inflow
  - wall_boundary_condition: bc_front_wall
  - open_boundary_condition: bc_right_open
  - symmetry_boundary_condition: bc_top
  - nonconformal_boundary_condition: bc_nc
```

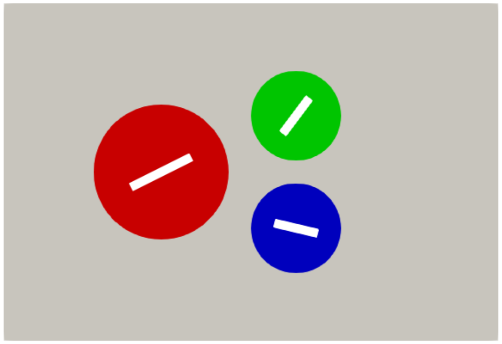
```
material_properties:

  target_name: block_1

  specifications:

    - name: density
      type: constant
      value: 1.0

    - name: viscosity
      type: constant
      value: 1.8e-5
```

```
material_properties:

  target_name: block_1

  specifications:

    - name: density
      type: ideal_gas

    - name: viscosity
      type: polynomial
      coefficient_declaration:
```

```
- inflow_boundary_condition: bc_left
    target_name: surface_1
    inflow_user_data:
      velocity: [0.5,0.0,0.0]
      mixture_fraction: 0.0

- wall_boundary_condition: bc_back
    target_name: surface_7
    wall_user_data:
      user_function_name:
        velocity: wind_energy
      user_function_string_parameters:
        velocity: [mmTop_ss7]
      mixture_fraction: 1.0
```

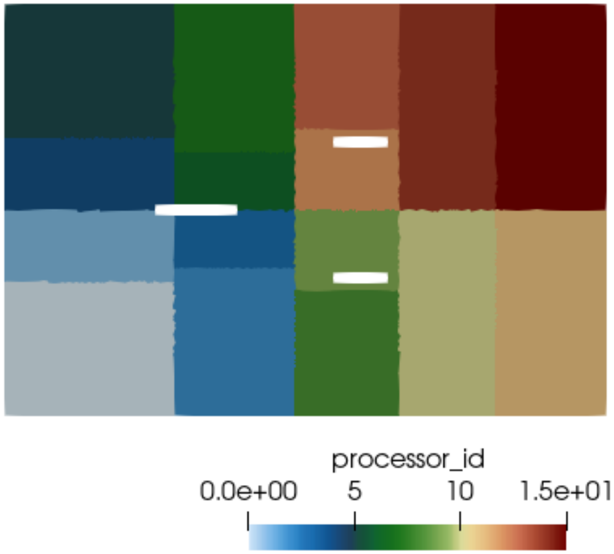
# A Note on Parallel Decomposition

- Decomposing the mesh into small subsets and operating on these subsets in parallel provides a methodology for increased simulation time



Colored by block ID

Colored by core count (= 1) One mpi rank.




16 MPI ranks = 16x faster (hopefully)

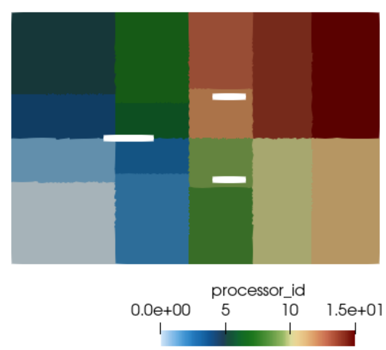
# How to Run in Parallel: Mesh Decomposition

- Given a mesh, mesh.g and myInputFile.i, if one desires to run on, e.g., 16 core case:
  1. Use pre-processing “decomposition” tools (Trilinos/install/bin)
  2. In-situ decomposition

myMesh.g

>/path/to/decomp –j {np} {meshName}
>mesh.g.16.00, mesh.g.16.01, .., mesh.g.16.15

1: - name: realmFluids
mesh: mesh/myMesh.g
~~automatic_decomposition_type: rcb~~



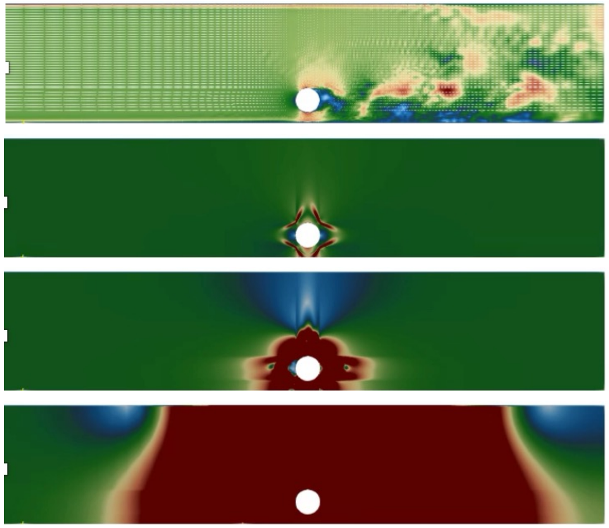

2: - name: realmFluids
mesh: mesh/myMesh.g
automatic_decomposition_type: rcb

>mpirun --np 16 /naluPath/naluX -i input.i &

>results.e.16.00, results.e.16.01, .., results.e.16.15
>input.log

# A Note on Divergence

- Whether by design or poor code usage, solution results from CFD analysis for complex applications may diverge. How do I know?
- Non-finite residuals, i.e., not-a-number:“nan” (either linear or non-linear),
  - const double norm = 0.0;
  - const double flux = rho*u*A/norm;
- What if the simulation runs, however, results look very wrong? Typical causes?
  - Bad initial condition that drives nonlinear solution to diverge
  - Too large of an initial time step
  - Poor stabilization, numerical parameters, etc.
  - Poor time integration
  - Ph.D. from ICME explaining why…



EBVC flow-past 2D cylinder
Top: +NOC
Bottom se: -NOC

# Test Case Input Files

- Input files are part of the Nalu regression test suite: Nalu/reg_test/test_files
- Mesh files are found under: Nalu/reg_test/mesh
- Formally, /mesh is a git submodule

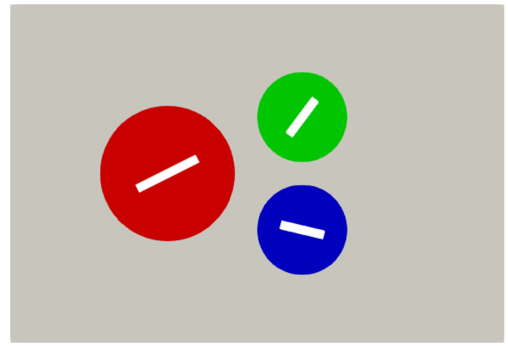
Test Cases Highlighted:

1. Nalu/reg_tests/test_files/dgNonConformalThreeBlade
2. Nalu/reg_tests/test_files/fluidsPmrChtPeriodic

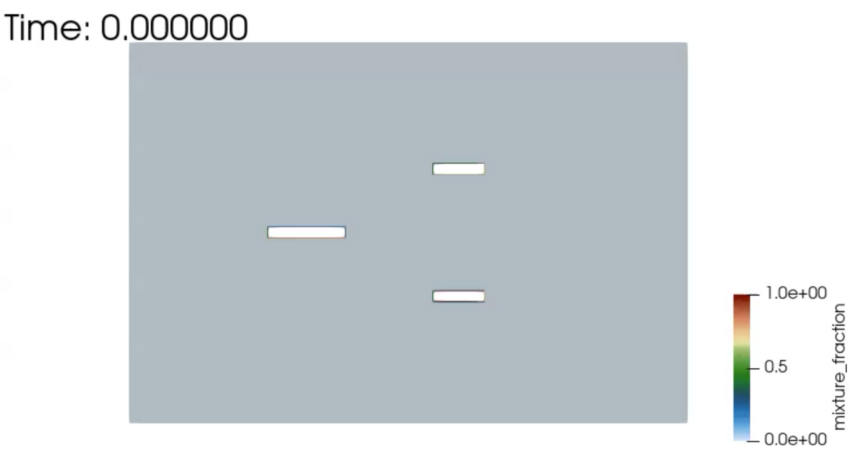
Resource: https://nalu.readthedocs.io/en/latest/source/theory/index.html

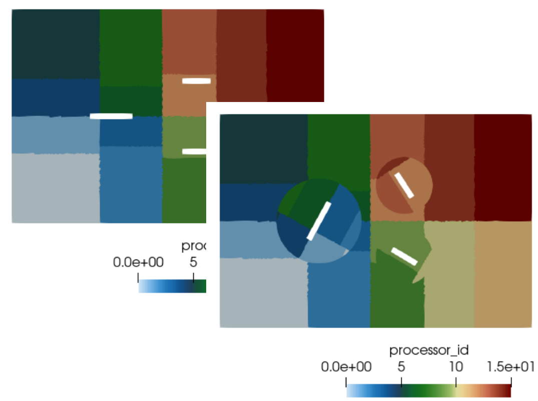
# Nalu/reg_tests/test_files/dgNonConformalThreeBlade

- Physics:
  - Flow past rotating square blades (Re = 10,000)
- Models
  - Newtonian fluid (air) with constant properties
- Boundary Conditions
  - Inflow, open, symmetry, DG/CVFEM interface



Domino, JCP, 2018




>mpirun --np 4 /naluPath/naluX -i dgNonConformalThreeBlade.i &

# Nalu/reg_tests/test_files/dgNonConformalThreeBlade

$U\Delta t/\Delta x$

Nonlinear iterations

Equations

Review

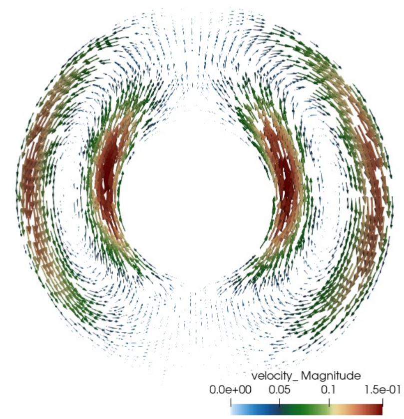
$\gamma_i$ for BDF2

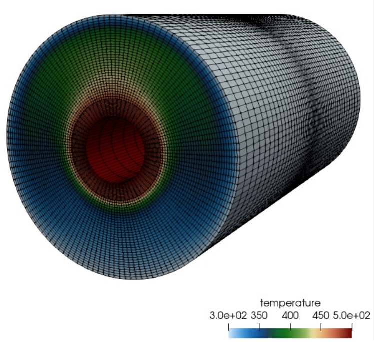
$\rho\ U\ L\ /\ \mu$

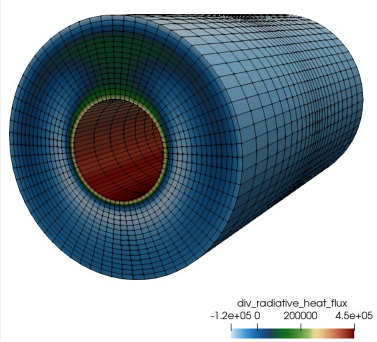
```
Time Step Count: 43 Current Time: 0.108355
 dtN: 0.002285 dtNm1: 0.00228982 gammas: 1.49947 -1.99789 0.498422
 Volume  0.818029 min: 1.34303e-06 max: 7.25544e-05
 Max Courant: 2.00422 Max Reynolds: 233.316 (realm_1)
   Realm Nonlinear Iteration: 1/1

realm_1::advance_time_step()
NLI    Name           Linear Iter      Linear Res     NLinear Res     Scaled NLR
---    ----           -----------      ----------     -----------     ----------
1/2    Equation System Iteration
 1/1      myLowMach
        MomentumEQS             5      1.36481e-07    0.000150157              1
        ContinuityEQS          10      9.02089e-07    0.000591053              1
        PNGradPEQS             28      2.69559e-09    1.25557e-06              1
        PNGradUEQS             30      6.93157e-09    3.99347e-06              1
 1/1            myZ
        MixtureFractionEQS      5      7.91662e-09    7.70262e-06              1
        PNGradZEQS             14      3.76375e-09    3.75761e-06              1
2/2    Equation System Iteration
 1/1      myLowMach
        MomentumEQS             5      3.54267e-09    1.77926e-06      0.0118493
        ContinuityEQS           9      6.32285e-07    0.000379046       0.641307
        PNGradPEQS             26       2.3035e-09    1.11406e-06       0.887296
        PNGradUEQS             26       5.6359e-10    3.39699e-07      0.0850636
 1/1            myZ
        MixtureFractionEQS      4      3.36865e-09    1.60783e-06       0.208738
mixFrac clipped (-) 14294 times; min: -0.0414158
mixFrac clipped (+) 2 times; max: 1.03823
        PNGradZEQS             18      1.32816e-10     6.6934e-08      0.0178129
Mass Balance Review:
Density accumulation: 0
Integrated inflow:     -0.375
Integrated open:       0.3749994376540416
Total mass closure:    -5.62346e-07
Mean System Norm: 6.404895358841074e-05 43 0.108355
```

# Nalu/reg_tests/test_files/fluidsPmrChtPeriodic

- Physics:
  - Uniformly emitting/absorbing participating media radiation (PMR) conjugate heat transfer (CHT) with buoyancy
- Models:
  - Newtonian fluid (air): ideal gas
- Boundary Conditions:
  - Wall, periodic




Stark#, cond/rad

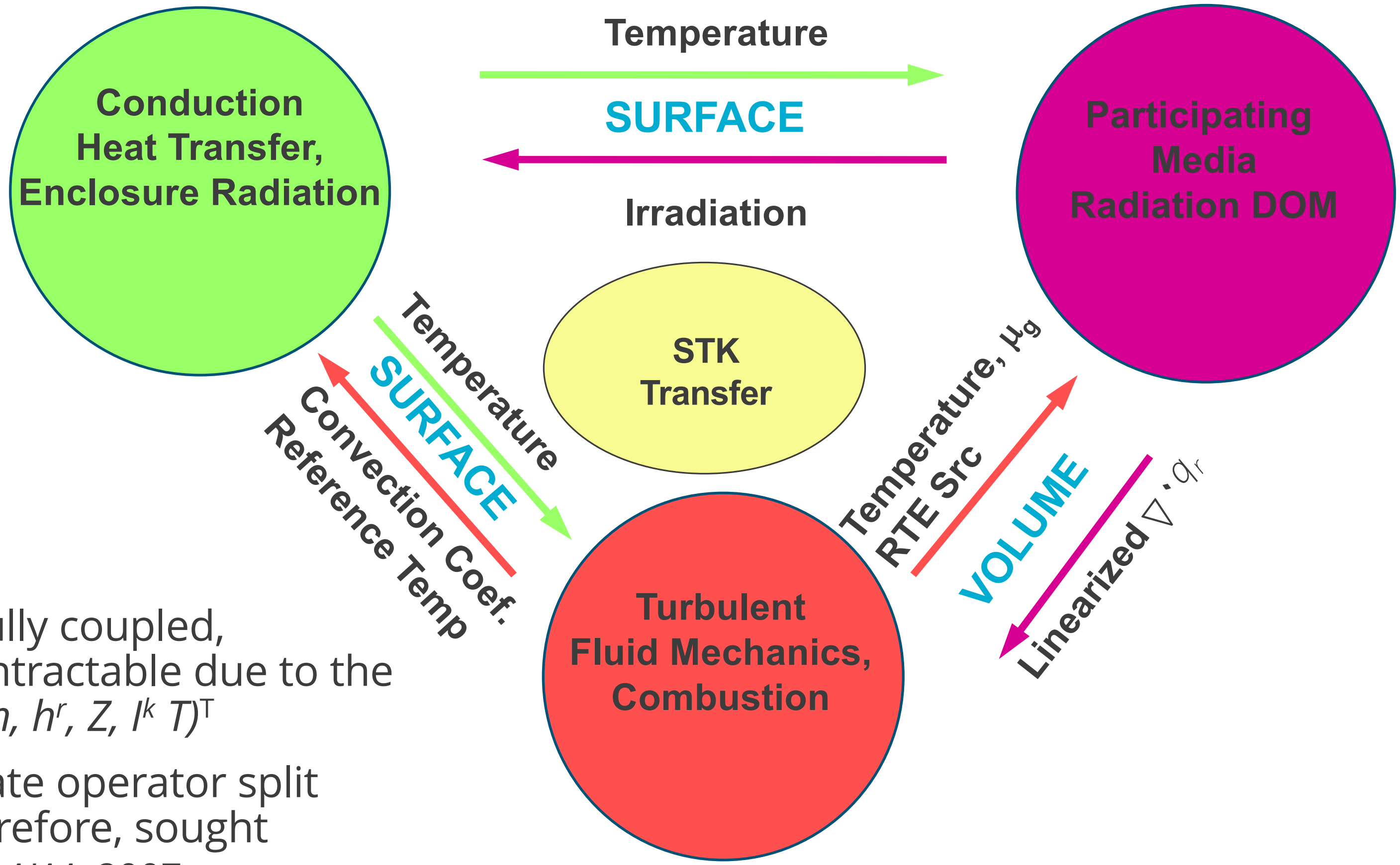
$Sk = \lambda\ \mu / \sigma\ T_i^3 \sim 0.4$

Rayleigh#, $\tau^{ThermalDiff} / \tau^{Conv}$

$Ra = g\ \beta\ (T_i - T_o)\ L / Pr\ \alpha^2 \sim 2e6$

>mpirun --np 8 /naluPath/naluX -i fluidsPmrChtPeriodic.i &

## Nalu/reg_tests/test_files/fluidsPmrChtPeriodic




- Monolithic, i.e., fully coupled, approaches are intractable due to the size of ($u, v, w, p, h, h^r, Z, I^k\ T)^T$
- Stable and accurate operator split methods are, therefore, sought
  - See Domino et al., *AIAA*, 2007

# Break-out Example Using Paraview

- Live demo